SECTION II—REMARKS

Applicants thank the Examiner for a thorough review, and respectfully request reconsideration of the above referenced patent application for the following reasons:

Claims 1-37 rejected under 35 U.S.C. § 112

The Office Action rejected claims 1-37 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. In particular, the Office Action objects to the "generating multiple views…" limitation of independent claim 1, as a mere repetition of the "generating a plurality of Web service definitions…" limitation, also of claim 1.

Applicants respectfully submit that claims 1-37 are canceled herein without prejudice, and thus, the rejection of claims 1-37 is rendered moot. However, Applicants present new claims 38-74 herein, which do not recite "generating multiple views," as objected to by the Examiner. Although similar limitations are claimed, Applicants have drafted new claims 38-74 an a manner that more clearly delineates the differences between "multiple views" and multiple "Web service definitions," each of which have different characteristics, as taught by Applicants and as claimed herein.

Accordingly, Applicants respectfully request the Examiner to withdraw the rejection to claims 1-37 and allow new claims 38-74 presented herein.

Claims 1-10, 17-30 and 33-35 rejected under 35 U.S.C. § 103(a)

The Office Action rejected claims 1-10, 17-30 and 33-35 under 35 U.S.C. § 103(a) as

being unpatentable over U.S. Patent Application Publication No. 2004/0205104 to Harvey et al.

("Harvey") in view of "Understanding Web Services: *XML, WSDL, SOAP and UDDI*," by Eric

Newcomer ("Newcomer"). Applicants respectfully submit that claims 1-37 are canceled herein

without prejudice, and thus, the rejection to claims 1-10, 17-30 and 33-35 is rendered moot.

However, Applicants respectfully submit that new claims 38-74 presented herein are patentable

over the prior art of record and in condition for allowance. For example, new independent claim

38 recites in pertinent part:

> A computer-implemented method for generating a deployable Web service archive, comprising:
> selecting a Web service implementation comprising a plurality of Web service operations and a plurality of Web service parameters;
> **generating a first virtual interface** to the Web service implementation, the first virtual interface to **expose a first subset** of the Web service operations and Web service parameters;
> **generating a second virtual interface** … to **expose a second subset** of the Web service operations and Web service parameters **different**, at least in part, **than the first subset** of the Web service operations and Web service parameters …
> generating a Web service definition …
> generating a Web service deployment descriptor … and
> generating the deployable Web service archive … .

Description of the claimed limitations:

In an effort to expedite prosecution of the application, Applicants provide a brief

overview of some novel aspects of the claimed limitations.

Applicants teach in the specification, methods for generating multiple Web services from

a single underlying Web service implementation through the use of multi-layer abstraction and

further packaging the Web services in a deployable Web services archive. More particularly,

Applicants teach a novel method to enable a single Web service implementation, including its underlying code (e.g., business logic), operations, parameters, naming conventions, data types, communication and authentication implementations, to be represented as multiple different Web services, each having potentially unique combinations of functionality, communication and authentication schemes, parameters, callable methods, etc.

Various Web services clients may have different capabilities for communicating with a Web service, for example, they may operate on different communication protocols, support different native data types, or provide varied levels of authentication. Using traditional Web services, a single Web service implementation (e.g., business logic, functionality, communication protocols, authentication levels, data types, naming conventions, etc.) must be programmed at development time for every potential Web services client to be supported, resulting in numerous individual Web service implementations that must be written, debugged, coordinated, and maintained.

However, through practice of the disclosed invention, a single Web service implementation can be written, and through practice of the disclosed multi-layer abstraction techniques, multiple Web services can be generated, packaged, deployed, and even listed individually as separate Web services with a UDDI directory. Each generated Web service can support diverse protocols, data types, naming conventions, and functionality, without modification to the underlying Web service implementation.

Consider for example, excerpts from Applicants' specification as originally filed, teaching in pertinent part:

> [0007] In some cases, it may be advantages to provide a Web service archive that **describes the Web service in terms of abstract layers**. For example, an abstract description of the Web service might describe features of the Web service in a **first layer**

of **abstraction** and protocol implementations of the features in a **second layer of abstraction**. Describing the Web service in multiple layers of abstraction provides for a flexible Web service framework. Conventional Web service archives **do not provide** Web service descriptions with multiple layers of abstraction.

* * *

[00024] An advantage to the architecture of Web service 400 is that **a single implementation** 410 **may be exposed in multiple ways**. ... conventional[] Web services generate separate implementations based on a single WSDL document.

Thus, Applicants' specification contemplates "Web service archives" having a multi-layer abstraction framework capable of exposing a single Web service implementation in multiple ways. Applicants further teach:

[00018] **Web service implementation** 410 **is the actual logic behind Web service** 400. In an embodiment, enterprise session bean 412 is the logic that provides the methods of Web service 400. The term "enterprise bean" refers to business logic that retrieves and/or processes data and provides that data to, for example, a user. ... .

* * *

[00020] ... Virtual interface ... **selectively exposes methods and parameters** of Web service implementation 410.

Thus, Applicants teach and claim "a Web service implementation comprising a plurality of Web service operations and a plurality of Web service parameters." Applicants further teach:

[00093] ... a **virtual interface** is an **interface for a Web service implementation**. In an embodiment, **one or more** virtual interfaces may be defined for the Web service implementation. Each virtual interface **may be published** to a service directory **separately**. A virtual interface may be used to, for example, **hide (or expose) operations and/or parameters**, change the names of operations and/or parameters, set standard (or default) values for parameters, convert parameter data types, and/or define how a parameter is represented ... .

[00094] ... The name of the operation may be changed [and] ... characteristics of a parameter such as its name, data type, default value, and/or whether it is exposed may be defined. ... .

Thus, Applicants teach "a first virtual interface to the Web service implementation, the first virtual interface to expose a first subset of the Web service operations and Web service parameters," as claimed. The passage further teaches that more than one virtual interface may be defined for a single Web service implementation, and that each virtual interface is publishable as a separate distinct Web service to a service directory.

The specification as originally filed further teaches, "generating a **Web service definition** for each of the first and second virtual interfaces," as claimed. For example:

> [00022] ... the capabilities of and requirements of Web service 400 are described in terms of **abstract features and properties** in Web service definition 424. ... there may be multiple Web service definitions 424 for each virtual interface 422.
> * * *
> [00095] ... a Web service definition is created to specify a behavior of the defined virtual interface. In an embodiment, features such as **communication type** or **authentication level** are assigned in abstract form in the Web service definition ... the technical details to implement the features may be provided in a Web service deployment descriptor.

Lastly, the specification teaches, "generating a **Web service deployment descriptor** for each Web service definition." For example:

> [00098] ... the **Web service deployment descriptor** specifies the **technical implementations** of the features that are described in the Web service definition. ... .
> [00099] ... A **transport binding** for Web service deployment descriptor 2700 is specified at 2720. In an embodiment, transport binding 2720 may include, for example, HTTP, FTP, SMTP, SOAP over HTTP, SOAP over FTP, SOAP over SMTP, and the like. In an embodiment, an **authentication protocol** is specified for deployment descriptor ... .
> * * *
> [00043] ... In an embodiment, three levels of authentication are supported: **none, basic, and strong**. The basic level of authentication may include authentication of a user name and password. The strong level of authentication may include client certificates to validate a message. ... .
> * * *

At "Table 2" of the specification, following paragraph [**00071**], several authentication types are listed, including: "HTTP with user name and password," "HTTP secured through the Secure Socket Layer," and "X.509 Client Certificates using HTTP secured through SSL."

Thus, Applicants teach and claim novel methods for enabling a single Web service implementation to support multiple and diverse Web services through multi-layer abstraction.

Brief description of Harvey:

Applicants provide a brief overview of the primary reference cited in the Office Action, specifically "Harvey." Harvey attempts to solve a different problem than Applicants, and thus, raises considerable confusion when it is used as a basis for rejecting various limitations claimed by Applicants.

In particular, Harvey does not contemplate or disclose mechanisms for enabling a single Web service implementation, including the business logic, parameters, and so forth, to support multiple and separately publishable Web services. Instead, Harvey focuses on the Web services directories, such as UDDI directories, where the Web services are eventually published.

The difference is stark, as Applicants teach and claim methods for generating deployable Web services archives that are **published to** a web services repository, and Harvey focuses on the management, searching, listing, and accessing of already published Web services **from** a UDDI directory structure. In other words, Applicants teach and claim methods for more efficiently **creating** Web services, all of which is "upstream" of the UDDI directory or prior to publishing to a UDDI directory, while Harvey, in contrast, discloses mechanism for improving accessing, using, and managing Web services and the UDDI directory itself, all of which comes after or "downstream" from the creation of any particular Web service.

Because Applicants teach methods upstream from a UDDI directory server and Harvey discloses mechanisms implemented downstream from a UDDI directory server, the subject matter of each will not overlap in any meaningful way. Much of the terminology may seem related, but each is discussing fundamentally different concepts.

To help alleviate any past confusion, Applicants have drafted new claims in an attempt to clarify the claimed subject matter, and in an effort to seek as expeditious allowance of the application as possible.

Harvey and Newcomer fail to disclose the claimed limitations:

Applicants respectfully submit that Harvey and Newcomer, whether considered alone or in combination, fail to disclose the limitations of new claim 38. For example, Applicants recite:

> generating **a first virtual interface** to the Web service implementation, the first virtual interface **to expose a first subset** of the Web service operations and Web service parameters;
> generating **a second virtual interface** … **to expose a second subset** of the Web service operations and Web service parameters **different**, at least in part, **than the first subset** of the Web service operations and Web service parameters …

In its rejection of now canceled independent claim 1, the Office Action states that Harvey discloses, "abstracting methods and parameters of a Web service implementation to a virtual interface." Refer to the Office Action at page 5, paragraph 14.

In support of its assertion, the Office Action refers to Harvey at paragraphs [0065], [0088], [0145], and [0277], generally describing that, "TModels defined by a user can be made children of the user object," which, according to Harvey, "enhances manageability and security" by allowing the user to "only modify and/or control their own sub-tree." Refer to Harvey, paragraph [**0065**]. Harvey goes on to describe various desired characteristics of a UDDI directory, for example, Harvey states that, "UDDI **should meet** performance and reliability

requirements" and that "[p]erformance **should not** suffer as the registry grows." Harvey further states that "the UDDI Registry **should be** industrial strength and fully support transactions and automatic recovery," and that "UDDI servers **should have** a high degree of availability" and "[s]ystem administrators **should have** capabilities to make the UDDI registry easy to maintain, monitor and control." Refer to Harvey, paragraph **[0088]**. Paragraph **[0145]** states that the UDDI standard does not "dictate[] how the user information is stored," and paragraph **[0277]** states that making tModels "children" of a user object makes security easier to implement.

While Harvey does disclose various desirable characteristics of a UDDI directory, and further emphasizes that making tModels children of a user object improves security of the UDDI directory, Harvey is utterly silent with regard to a "virtual interface … **to expose a first subset** of the Web service **operations** and Web service **parameters**" belonging to a particular Web service implementation.

The Office Action appears to equate the "tModels" disclosed by Harvey to the "virtual interface" claimed by Applicant. For example, the Office Action states, "[t]he stored 'tModels' allow control of specific implementations of a [W]eb service by particular users." Refer to the Office Action at page 5, paragraph 14.

A "tModel" is not an interface to a Web service implementation. Rather, a tModel is meta-data that allows UDDI directories to provide searchable descriptions of available Web services. Consider for Example, Harvey's own definition:

> **[0240]** A TModel represents an idea. That idea might be, for example, **a categorization system, requiring the specification of values which may be validated**. Or it may be a specification of a data communication protocol. TModels are a flexible and powerful concept, and central to the **ability of UDDI to represent complex data in a way that can be <u>accurately queried</u>**.
> **[0241]** The only required elements of the TModel object are a TModel key and a name. These are represented as strings.

* * *

>[0297] ... TModels are used throughout UDDI for various purposes. **They are _categorization keys_, search identifiers, (UDDI) relationship descriptors**, and in this instance, technical specification `fingerprints`. A TModel which is `attached` to a BindingTemplate describes a technical specification to which that BindingTemplate (see FIG. 8) conforms. ... .

Thus, Harvey describes tModels as a categorization tool, to aid in the representation of "complex data in a way that can be accurately queried." This definition is inconsistent with the Office Action's assertion that a tModel is equivalent to a "virtual interface."

Even if, in arguendo, Harvey's tModel were equivalent to a "virtual interface," which Applicants do not concede, Harvey does not disclose that the tModels "expose" anything, much less "**expose a _first subset_** of the Web service operations and Web service parameters." Moreover, Harvey does not disclose that a "second virtual interface" exposes "**a _second subset_** of the Web service operations and Web service parameters **_different ... than the first subset_**."

And further still, equating Harvey's tModels to Applicants' "virtual interface" teaches away from Applicants, as Harvey specifies in the above passage that tModels contain "technical specification 'fingerprints'" which include, for example, "a specification of a **data communication protocol**" which is the opposite of what Applicants teach and claim. Applicants teach and claim that a "communication protocol" is the "Web service deployment descriptor" that "defin[es] a **communication protocol to implement** the specified protocol-independent communication type." Refer to new claim 38.

Newcomer fails to cure the deficiencies of Harvey, as it too fails to disclose "a first virtual interface ... to expose a first subset of the Web service operations and Web service parameters" or a "second virtual interface ... to expose a second subset ... different ... than the first subset."

Because the combination of Harvey and Newcomer, whether considered alone or in combination, fails to disclose at least one limitation that Applicants recite in new independent claim 38, Applicants respectfully submit that claim 38 is patentable over the references and is in condition for allowance. Applicants further submit that independent claims 51, 60, and 69, which recite similar limitations, as well as those claims depending on independent claims 38, 51, 60, and are patentable over the references and in condition for allowance.

Accordingly, Applicants respectfully request the Examiner to withdraw the rejection to claims 1-10, 17-30 and 33-35 and allow new claims 38-74.

Dependent claims 11-16, 31, 32, 36 and 37 rejected under 35 U.S.C. § 103(a)

The Office Action rejected claims 11-16, 31, 32, 36 and 37 under 35 U.S.C. § 103(a) as being unpatentable over Harvey in view of Newcomer, and further in view of "BEA WebLogic Server 6.1" ("BEA1") and web.xml Deployment Descriptor Elements" ("BEA2").

Applicants respectfully submit that claims 1-37 are canceled herein without prejudice, and thus, the rejection of dependent claims 11-16, 31, 32, 36 and 37 is rendered moot. However, Applicants respectfully submit that new claims 38-74 presented herein are patentable over the prior art of record and in condition for allowance for the reasons discussed above with regard to the independent claims rejected under 35 U.S.C. § 103(a). In particular, the additional references of BEA1 and BEA2, whether considered alone or in combination with Harvey and Newcomer fail to cure the deficiencies of Harvey as neither discloses "a first virtual interface ... to expose a first subset of the Web service operations and Web service parameters" or a "second virtual interface ... to expose a second subset ... different ... than the first subset."

Accordingly, Applicants respectfully request the Examiner to withdraw the rejection to claims 11-16, 31, 32, 36 and 37 and allow new claims 38-74.

New claims 38-74

In accordance with the preceding remarks, Applicants respectfully submit that new claims 38-74 are patentable over the prior art of record and in condition for allowance. New claims 38-74 find support in the specification as originally filed and in the original claims.

Accordingly, Applicants respectfully request the Examiner to allow new claims 38-74 presented herein.

CONCLUSION

Given the above amendments and accompanying remarks, all claims pending in the application are in condition for allowance. If the undersigned attorney has overlooked subject matter in any of the cited references that is relevant to allowance of the claims, the Examiner is requested to specifically point out where such subject matter may be found. Further, if there are any informalities or questions that can be addressed via telephone, the Examiner is encouraged to contact the undersigned attorney at (503) 439-8778.

Charge Deposit Account

Please charge our Deposit Account No. 02-2666 for any additional fee(s) that may be due in this matter, and please credit the same deposit account for any overpayment.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

/Gregory D. Caldwell/_____   Date: June 25, 2008
Gregory D. Caldwell
Registration No. 39,926
Attorney for Applicants

Blakely, Sokoloff, Taylor & Zafman LLP
1279 Oakmead Parkway
Sunnyvale, CA 94085-4040
Telephone: (503) 439-8778
Facsimile: (503) 439-6073